



TITLE:

辺上を移動するロボット1台による
最適な多角形探索 (理論計算機科学
の深化: 新たな計算世界観を求めて
)

AUTHOR(S):

深見, 浩和; 小野, 廣隆; 定兼, 邦彦; 山下, 雅史

CITATION:

深見, 浩和 ...[et al]. 辺上を移動するロボット1台による最適な多角形探索 (理論計算機科学の深化: 新たな計算世界観を求めて). 数理解析研究所講究録 2008, 1599: 182-188

ISSUE DATE:

2008-05

URL:

<http://hdl.handle.net/2433/81777>

RIGHT:

辺上を移動するロボット 1 台による最適な多角形探索

深見 浩和* 小野 廣隆† 定兼 邦彦† 山下 雅史†

* 九州大学大学院システム情報科学府

† 九州大学大学院システム情報科学研究院

1 はじめに

商業施設など、多角形状の施設はたくさんある。このような施設に侵入者を発見するロボットを導入することを考える。ロボットはサーチライトを装備しており、移動することができるものとする。セキュリティ上の観点から、もし侵入者が存在したとすると、その動きにかかわらず確実に発見できるようにしたいという要望が生じる、では探索ロボットとそのライトをどのように移動させれば良いだろうか。この問題は多角形探索問題と呼ばれ、[2] で防犯カメラを用いる美術館問題 [1] の動的拡張として導入された。ロボットが装備しているサーチライトの本数や、ロボットの台数・能力などの設定によりさまざまな探索モデルが考えられており、ロボットの能力 (モデル) によっては侵入者が内部をずっと逃げ回ることが可能な多角形も存在する。本研究では用いるロボットの台数を 1 台とし、(i) ライトを 1 本持ち、(ii) 境界上のみ (多角形の辺上のみ) を移動することができる、という制約の下、探索可能な図形に対する考察を行う。

本文の構成は以下の通りである。まず、2 節において扱う探索モデルに関する定義を行い、VOD と呼ばれる図に関する定義を行う。3 節ではロボット 1 台による探索可能性について述べ、その後 4 節で総移動量を最小化することに関してを述べる。5 節でライトの移動量を最小化することに関して述べた後、最後に 6 節でまとめと今後の課題について述べる。

2 準備

2.1 モデル

多角形 P は内部に穴の空いていないものとし、二次元平面上に描かれているものとする。多角形の辺

集合を境界と呼び、 ∂P で表す。多角形の周の長さを $|\partial P|$ で表し、 $|\partial P| = 1$ とする。辺上の点を時計回りに順序付け、辺上の点 a から、周に沿って時計回りに進み点 b にたどりつくまでの開区間を $\partial P(a, b)$ で表し、閉区間を $\partial P[a, b]$ で表す。

探索ロボットと侵入者は二次元平面上の点で表される。各ロボットはライトを 1 本持っており、その光は向けている方向のみに進み、境界によって遮られるものとする。ライトの光が境界にぶつかった点とロボットの位置を結ぶ線分を視線と呼ぶ。視線上に侵入者が存在する時侵入者を発見したとする。

侵入者は多角形内部を任意の速度で移動することができるものとする。また侵入者は視線上を通過すると発見されてしまうため、多角形内部を視線上を通過しないように移動するものとする。

2.2 定義

侵入者は視線にぶつかるか発見されてしまうため、ロボットの位置とライトの先端の位置が一致した状態から探索を始めた場合、ある時刻において侵入者がいないことが保証される領域が存在する。この領域をクリア領域と呼ぶ。これとは逆に侵入者が存在しうる領域を非クリア領域と呼ぶ。

定義より、すべての領域をクリア領域にできた時、侵入者は発見されている。侵入者発見のためのロボット・ライトの先端の移動方法 (スケジュールと呼ぶ) が存在する場合、その多角形は探索可能であると呼ぶ。

本研究で扱う多角形の中には凹多角形も含まれ、内角が 180° を越える頂点が存在するものもある。この頂点を **reflex vertex** と呼び、この頂点の存在によって周上のある点から直接ライトで照らすことのできない点が存在する。また、視線が **reflex**

vertex に接している場合は遮られていないとする。

ロボットが止まった状態でライトを回転させると、視線を遮る辺が無い場合ライトの先端は辺上を連続的に移動する。しかし回転中に視線が reflex vertex にぶつかる、もしくは逆にライトの先端が reflex vertex にたどり着き、視線を遮るものが無くなると、ライトの先端は辺上を非連続的に移動する。これをジャンプと呼ぶ。

頂点 p_i, p_{j+1} が非 reflex vertex であり、 p_{i+1} から p_j までがすべて reflex vertex である領域を Concave と呼び、 $C = (p_i, p_{j+1})$ で表す。定義より 2 つの Concave は共通部分を持たないことがわかる。

頂点番号の小さい順に Concave に順番を付け、 k 番目の Concave を $C_k = (p_i, p_{j+1})$ とする。 p_i から p_{i+1} の方向に線分を伸ばし、最初に辺にぶつかる点を a'' 、 p_{j+1} から p_j の方向に線分を伸ばし、最初に辺にぶつかる点を b'' とする。この 2 点に p_i, p_{j+1} を加えた 4 点を RedPoint と呼ぶ。この定義から、Concave を m 個もつ多角形は RedPoint を高々 $4m$ 個持つことがいえる。

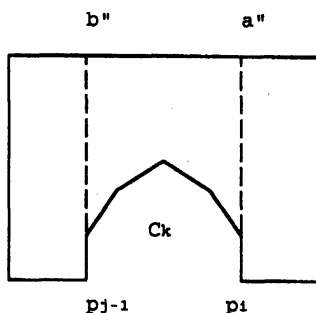


図 1: Concave

2.3 VOD

reflex vertex の存在により、境界上のある点から直接照らすことのできない点が存在することから、境界上の任意の 2 点 a, b の関係は、

- a から b を直接照らすことができる
- a からは遮る辺があり直接照らすことはできない
- a と b は同じ辺上である

の 3 つに分類できる。本研究ではこの関係を図にした VOD (visibility Obstruction diagram) と呼

ばれる図を用いる。図の縦軸はロボットの多角形上での位置を表し、横軸はライトの先端の位置を表す。

境界上の点 a から境界上の点 b を照らすことを考える。視線が多角形の内部のみを通り、遮る辺が無い場合は VOD 上の点 (a, b) を白、外部を通る場合や遮る辺がある場合は灰で塗り分ける (図 2)。また $y = x$ の部分を S 、 $y = x - 1$ の部分を G と呼ぶ。

VOD 上の点はロボットの位置とライトの先端の位置に対応するので、VOD 上での点の移動はロボットの移動・ライトの先端の移動に対応する。このことから、スケジュールは VOD 上の有向パスとなることがわかる [3]。

この有向パスの始点が S 上である場合を考える。 S 上の点はロボットの位置とライトの先端の位置が同じ位置にあるという状態を表しており、この点から移動を開始すると定義よりクリア領域が生じる。また移動を行い点 (b, a) にたどり着いたとすると、 $\partial P(a, b)$ と線分 \overline{ab} で囲まれる領域がクリア領域となることがわかる。これはロボットから見て視線の左側が常にクリアであることを意味するので、この状態を左不変と呼ぶ。これとは逆に、ロボットから見て視線の右側が常にクリアになっている状態を右不変と呼ぶ。

左不変を維持したまま G 上の点 (その点を $(m, m - 1)$ とする) にたどり着いたとすると、 $\partial P(m - 1, m)$ と線分 $\overline{(m - 1)m}$ で囲まれる領域がクリア領域となるため全ての領域がクリア領域となり、探索が完了することがわかる。もし、左不変を維持したまま探索を行うスケジュールで侵入者発見が可能であるとすると、向きを逆にしたスケジュールを実行すると右不変を維持したまま探索を行うスケジュールとなり、やはり侵入者発見が可能である。本研究では左不変を維持したまま探索を行うスケジュールのみを扱う。

次にジャンプについて考える。ライトの先端が時計回りの方向にジャンプすることは VOD 上では灰色部分を右向きに飛び越えることに対応し、反時計回りの方向にジャンプすることは VOD 上では灰色部分を左向きに飛び越えることに対応する。

VOD 上の上下のジャンプについて考える。これはロボットがライトを動かさずに reflex vertex から領域内部を通して辺上の点に移動する、もしくは逆の移動に対応する。しかし本研究のモデルより境界上を離れることはできないため、このジャンプは

生じない。

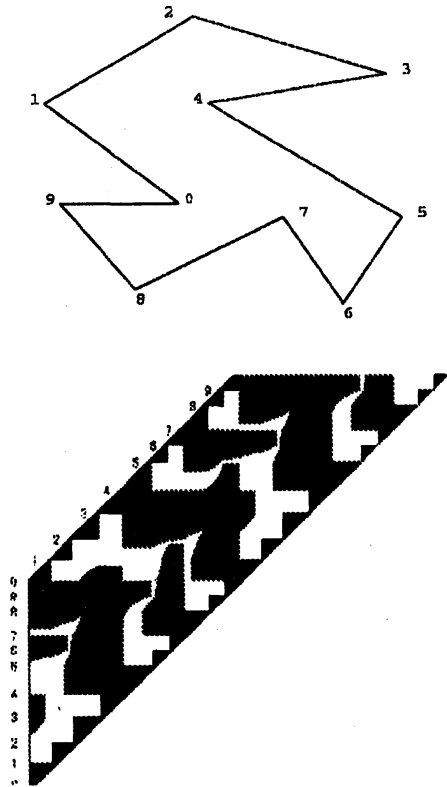


図 2: VOD の例

3 VODでの判定

与えられた多角形がロボット 1 台で探索可能であるかは、VODを用いて判定できることが [3] で示されている。

定理 1 与えられた多角形がロボット 1 台で探索可能である必要十分条件は、 S から G まで、右向きのジャンプを許す白い部分のみを移動するパスが存在することである。

3.1 Skeleton

S から G までのパスを見つける際、灰色の領域が探索の障壁となる。ここではその障壁を直線で近似することを考える。

- 対角線, $D = \{(p, p) | p \in \partial P\}$

- 水平線, $AH = \bigcup_{k=0}^{m-1} \{(q, a'_k) | q \in H(a'_k)\}$ と $BH = \bigcup_{k=0}^{m-1} \{(q, b'_k) | q \in H(b'_k)\}$
- 垂直線, $AV = \bigcup_{k=0}^{m-1} \{(a'_k, q) | q \in H(a'_k)\}$ と $BV = \bigcup_{k=0}^{m-1} \{(b'_k, q) | q \in H(b'_k)\}$

これを **Skeleton** と呼び、 $SK = D \cup AH \cup BH \cup AV \cup BV$ で表す。また VOD と同じく 2 次元平面状に描いたものを **Skeleton-VOD** と定義する。例として図 2 の多角形の Skeleton VOD を図 3 に示す。

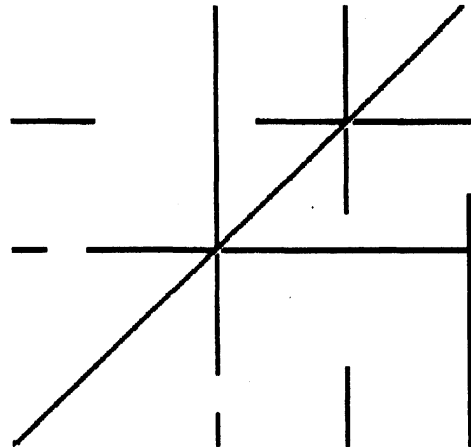


図 3: Skeleton VOD

この Skeleton-VOD 上でパスを発見することを考える。障壁の性質から、Skeleton-VOD を RedPoint でグリッドに区切ると、各マス目 (これをボックスと呼ぶ) の辺は Skeleton であるかそうでないかのどちらかになる。このことから、次のグラフ $G = (V, E)$ を作成する。

頂点は $V(G) = \{v_{i,j} | i, j \in Z_{4m}\}$ とする。 $v_{i,j}$ は左から i 番目、上から j 番目のボックスを表す。

辺は $E(G) = E^n - E^d - E^v - E^h$ とする。

- $E^n = \{(v_{i,j}, v_{i\pm 1, j}), (v_{i,j}, v_{i, j\pm 1}) | i, j \in Z_{4m}\}$
- $E^d = \{(v_{i,i+1}, v_{i,j}), (v_{i-1,i}, v_{i,j}), (v_{i,i}, v_{i+1,j}), (v_{i,i}, v_{i,j-1}) | i, j \in Z_{4m}\}$
- $E^v = \{(v_{i,j-1}, v_{i,j}), (v_{i,j}, v_{i,j-1}) | \exists k : ((r_j = a'_k) \wedge (r_j \in [a'_k, a''_k])) \vee ((r_j = b'_k) \wedge (r_j \in [b'_k, b''_k]))\}$
- $E^h = \{(v_{i-1,j}, v_{i,j}), (v_{i,j}, v_{i-1,j}) | \exists k : ((r_j = a'_k) \wedge (r_j \in [a'_k, a''_k])) \vee ((r_j = b'_k) \wedge (r_j \in [b'_k, b''_k]))\}$

このグラフ G で探索パスを発見することができれば、与えられた多角形は探索可能であることが [3] で示されている。

4 移動距離を最適化した探索

グラフ G で探索を行うことで、与えられた多角形を探索するスケジュールを得ることができるが、そこで得られたスケジュールがロボットの移動距離の面で最適であるとはいえない。ここでロボットの総移動量を最適とするスケジュールを発見することを考える。

(Skeleton)VOD 上の点の y 座標はロボットの辺上の位置を表すことから、(Skeleton)VOD 上で上下移動の総量を最小とするスケジュールを発見すればいいことがわかる。

まず、VOD 上のコスト最小パスに関して次の補題を示す。

補題 1 VOD 上の探索パス p がコスト最小の探索パスであるならば、始点の y 座標はいずれかの *RedPoint* である。

証明. 背理法で示す。 p の始点の y 座標が *RedPoint* で無いとする。このとき、始点から y 座標が変化するまでのパスを t_0 とすると、 t_0 は灰色の部分に接していないので t_0 を灰色部分に接するまで移動させることができる。移動後を t'_0 とすると、 $p' = t'_0 \cdot r_1 \dots$ は探索パスとなり、コストは p より小さいものとなるので仮定に矛盾する。よって示せた。 \square

系 1 VOD 上の探索パス p がコスト最小の探索パスであるならば、終点の y 座標はいずれかの *RedPoint* である。

ここで、グラフ G の上下方向の辺に重みとして移動コストを割り当てることを考える。しかしボックス内の位置がわからないので、各 $v_{i,j}$ をボックスの上側にいることを表す $v_{i,j}^u$ と、ボックスの下側にいることを表す $v_{i,j}^l$ の 2 つに分ける。この 2 点間に移動コストを割り当てたグラフ G_c を考える。

$$\begin{aligned} V(G_c) &= \{v_{i,j}^u, v_{i,j}^l | i, j \in Z_{4m}\} \\ E(G_c) &= \{(v_{i,j}^u, v_{i-1,j}^l) | (v_{i,j}, v_{i-1,j}) \in E(G)\} \cup \\ &\quad \{(v_{i,j}^l, v_{i+1,j}^u) | (v_{i,j}, v_{i+1,j}) \in E(G)\} \cup \end{aligned}$$

$$\begin{aligned} &\{(v_{i,j}^u, v_{i,j+1}^u) | (v_{i,j}, v_{i,j+1}) \in E(G)\} \cup \\ &\{(v_{i,j}^l, v_{i,j+1}^l) | (v_{i,j}, v_{i,j+1}) \in E(G)\} \cup \\ &\{(v_{i,j}^u, v_{i,j}^l), (v_{i,j}^l, v_{i,j}^u) | i, j \in Z_{4m}\} \\ w(e) &= \begin{cases} |\partial P[r_j, r_{j+1}]| & e = (v_{i,j}^u, v_{i,j}^l) \in E(G_c) \\ 0 & \text{others} \end{cases} \end{aligned}$$

このグラフ G_c 上で最短距離を求めるアルゴリズムを実行することで、移動量を最適化したスケジュールを得ることを考える。次節でコストが等しい VOD 上のパスに変換するアルゴリズムについて述べる。

4.1 変換アルゴリズム

ここではグラフ G_c 上の最短パス π を VOD 上のパスに変換することを考える。

Core

RedPoint の定義から、各ボックスには連結な白い領域を高々 1 つ含む。この領域を *Core* と定義する。また水平・垂直に切ったとき白い区間は高々 1 つである。また灰色領域の性質から、ボックス内で上側に灰色領域がある場合は境界の y 座標は x 座標が増加すると増加し、下側に灰色領域がある場合も境界の y 座標は x 座標が増加すると増加する。

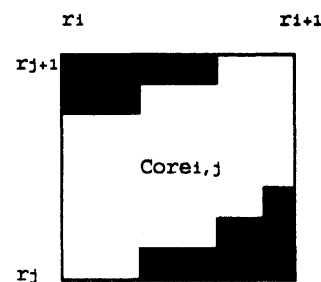


図 4: Core

変換アルゴリズム

- 入力: グラフ G_c 上でのパス $\pi = \tau_0 \rho_1 \tau_1 \dots \rho_k \tau_k$
- 出力: VOD 上のパス p .

ρ_i の形で場合わけを行う。 ρ_{i-1} で追加されたパスの終点を (x, y) とする。

1. $(v_{a,m}^l, v_{a,m-1}^u)$ のとき: $y \neq r_m$ ならば $Core_{a,m}$ の周囲を時計回りに $y = r_m$ となるところまで移動。
2. $(v_{a,m}^u, v_{a,m+1}^l)$ のとき: $y \neq r_{m+1}$ ならば $Core_{a,m}$ の周囲を時計回りに $y = r_{m+1}$ となるところまで移動。
3. $(v_{m,b}^u, v_{m+1,b}^u)$ のとき: $x \neq r_{m+1}$ ならば $Core_{m,b}$ の周囲を時計回りに $x = r_{m+1}$ となるところまで移動。
4. $(v_{m,b}^u, v_{m-1,b}^u)$ のとき: $x \neq r_m$ ならば $Core_{m,b}$ の周囲を反時計回りに $x = r_m$ となるところまで移動。
5. $(v_{m,b}^l, v_{m+1,b}^l)$ のとき: $x \neq r_{m+1}$ ならば $Core_{m,b}$ の周囲を反時計回りに $x = r_{m+1}$ となるところまで移動。
6. $(v_{m,b}^l, v_{m-1,b}^l)$ のとき: $x \neq r_m$ ならば $Core_{m,b}$ の周囲を時計回りに $x = r_m$ となるところまで移動。
7. $(v_{a,b}^l, v_{a,b}^u)$ のとき: 上に $Core_{a,b}$ の周囲にぶつかる所まで移動。
8. $(v_{a,b}^u, v_{a,b}^l)$ のとき: 下に $Core_{a,b}$ の周囲にぶつかる所まで移動。

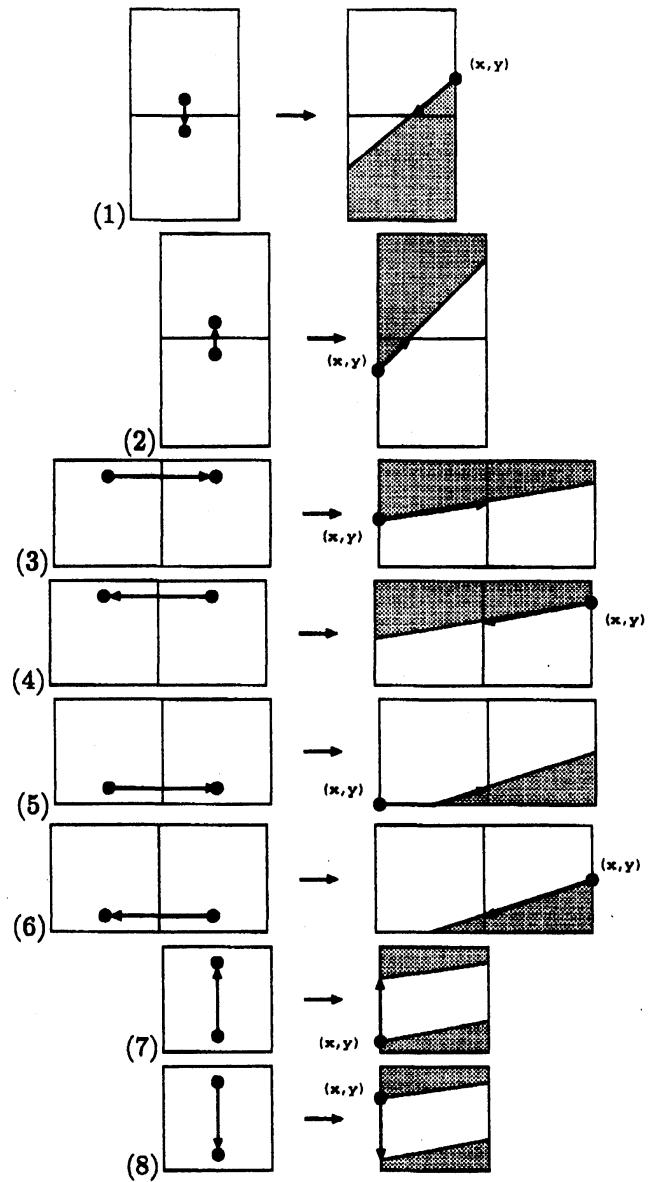


図 5: 変換アルゴリズム

4.2 正当性

本節では変換アルゴリズムの正当性について述べる。

補題 2 グラフ G_c 上のコスト最小な探索パス $\pi = \tau_0 \rho_1 \tau_1 \dots \rho_k \tau_k$ において, (7)(8) となっているものを $\rho_{\alpha_1}, \rho_{\alpha_2}, \dots, \rho_{\alpha_k}$ とする, すると ρ_{α_i} と $\rho_{\alpha_{i+1}}$ の間にかならず (1)(2) の移動が存在する。

証明. もし存在しないとすると, ρ_{α_i} と $\rho_{\alpha_{i+1}}$ の間には (3)~(6) のどれか 1 種類しか含まれず, ρ_{α_i} と $\rho_{\alpha_{i+1}}$ は逆向きとなる. この場合, 移動を行わないパスが存在するため, π が最短であるという事実に矛盾する. \square

定理 2 グラフ G_c 上の探索パス π に対して, コストの等しい VOD 上の探索パス p が存在する.

証明. 入力を $\pi = \tau_0 \rho_1 \tau_1 \dots \rho_k \tau_k$ とする. π 中でコストが生じるものは ρ_i が (7)(8) であるので, (7)(8) となっている各 ρ_i に対して, p でコストが等しい部分に置き換えられていることを示せばよい.

ここで (7)(8) で追加される辺のコストがグラフの辺のコストより小さい場合を考える. 補題 1 より前後に (1)(2) の移動が存在し, このときパスの端点は RedPoint の位置となることがいえる. また, 灰色部分の厚さは単調であることから, (7)(8) で追加される辺のコストと前後の部分を加えたコストはグラフの辺のコストと等しくなるといえる.

よって, グラフ G_c 上の探索パス π に対して, コストの等しい VOD 上の探索パス p が存在するといえる. \square

次に, VOD 上のパス p のコストより小さいグラフ G_c 上のパスが存在することを示す.

定理 3 VOD 上の探索パス p のコストを C_p とする. この時 C_p 以下のコストをもつ G_c 上の探索パス π が存在する.

証明.

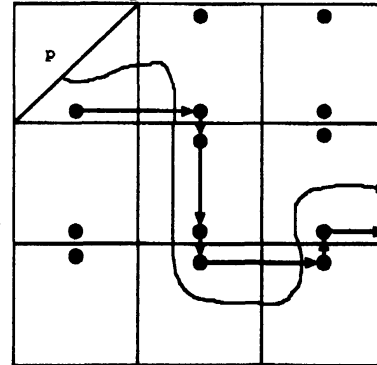
図 6 のように, VOD 上の探索パス p を, 通ったブロックと同じ順でたどる G_c 上のパス π に変換する. すると π でコストが生じるのは p がブロックを縦断したときであり, 定義から一方向に縦断した時 π で加える辺のコストと一致するので π のコストが p のコスト以下であるのは明らか. \square

4.3 提案アルゴリズム

前節から, ロボットの移動量を最小とするスケジューリング生成アルゴリズムを提案する.

1. P の頂点を入力する
2. RedPoint を求め, ソートする.

図 6: p から π への変換



3. RedPoint の情報から, グラフ G_c を作る
4. グラフ G_c 上で, 最短路を発見するグラフ探索アルゴリズムを用いて, 始点・終点がともに対角線上である最短路を見つける.
5. 変換アルゴリズムで VOD 上のパスを生成.

5 ライトの先端の移動量を最適化した探索

本節ではライトの先端の移動量をコストとした場合を考える. 基本的に前節と同じアイデアであるので, ここでは主な違いである左ジャンプの扱いについて述べる.

5.1 左ジャンプ

コストがライトの先端の移動量であることから, VOD では左右の移動量を最小化すればよい. しかしジャンプを行う際の y 座標によって必要となるコストは変化する. 飛び越える灰色領域は S から伸びるものと G から伸びるものの 2 種類があるので, 本節ではそれぞれに対しコスト最小なものを考える.

S から伸びるものは図 7 のようにボックスの上端でジャンプを行ったときコストは最小となる. よってジャンプ前のボックスの右端を表す点から, ジャンプ後のボックスの右端を表す点への辺をグラフ G_c に加え, 重みはジャンプによっていける一番上の高さから左に灰色領域にぶつかるまでの距離とする.

G から伸びるものは図8のようにボックスの下端でジャンプを行ったときコストは最小となる. よってジャンプ前のボックスの左端を表す点から, ジャンプ後のボックスの左端を表す点への辺をグラフ G_0 に加え, 重みはジャンプによっていける一番下の高さからジャンプを行った後, ジャンプ後のボックスの左端にぶつかるまでの距離とする.

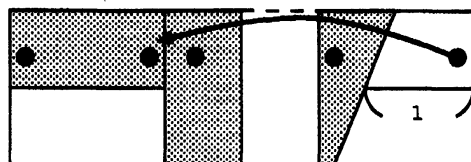


図 7: S から伸びる灰色をジャンプ

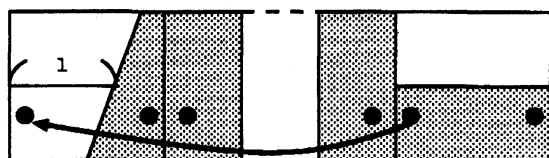


図 8: G から伸びる灰色をジャンプ

6 まとめと今後の課題

ライトを1本持った, 境界上のみを移動可能なロボット1台によるモデルでロボットの移動量を最小化したスケジュールを出力するアルゴリズムの提案を行った.

今後の課題として, ライトの先端の移動量ではなく, ライトの回転量も最小化したスケジュールを発見することと, ロボットの総移動量とライトのコストの和を最小化するスケジュールの発見があげられる.

参考文献

- [1] J.O'Rourke, "Art Gallery Theorems and Algorithms", Oxford University Press, New York, Oxford, 1987.
- [2] I.Suzuki and M.Yamashita, "Searching for a mobile intruder in a polygonal region", SIAM Journal on Computing, vol. 21,868-888, 1992.
- [3] S.M.LaValle, B.Simov and G.Slutski. "An algorithm for searching a polygonal region with a

flashlight", In Proceedings of the ACM symposium on Computational Geometry 2000, Hong Kong University of Science and Technology, 260-269, 2000